SISC: A Text Classification Approach Using Semi Supervised Subspace Clustering

Mohammad Salim Ahmed
Department of Computer Science
The University of Texas at Dallas
salimahmed@utdallas.edu

Latifur Khan
Department of Computer Science
The University of Texas at Dallas
lkhan@utdallas.edu

***Abstract*—Text classification poses some specific challenges such as high dimensionality with each document (data point) having only a very small subset of them and representing multiple labels at the same time. In this paper, we propose *SISC (Semi-supervised Impurity based Subspace Clustering)*, a κ-nearest neighbor approach, based on semi-supervised subspace clustering that considers all these factors during text classification. *SISC* finds clusters in the subspaces of the high dimensional text data. This novel semi-supervised subspace clustering algorithm is based on fuzzy cluster membership. This fuzzy clustering exploits chi square statistic of the dimensions and impurity measure within the cluster. Empirical evaluation on real world multi-class and multi-label data set reveals that our approach outperforms state-of-the-art text classification as well as subspace clustering algorithms.**

I. Introduction

Text classification is different from conventional classification approaches in two important aspects. The first one is in the construction of text documents. The dimensionality for text data is very large in comparison to other forms of data sets. Also each document may contain only a few of the features from the entire pool of feature set. The effect of this characteristic is that, distance measures based on TFIDF and euclidean distance usually do not perform well in the classification process [8]. The second difference from conventional classification is the presence of multiple labels associated with each document. This is due to the fact that a single document may cover multiple classes simultaneously. One approach may be to consider all class combinations and then run individual classification for each of them. But, considering all such combinations of classes present in such a multi-label dataset will make the classification infeasible [11], [13].

The notion of subspace clustering matches that of text data, i.e. having large dimensionality and possibility of each class to correspond to only a subset of features from the entire feature set. Subspace clustering allows us to find clusters in a weighted hyperspace [6] and can aid us in finding documents that form clusters in only a subset of dimensions. Each dimension of a subspace cluster contributes differently in forming those clusters. So, applying subspace clustering can, to a large degree, divide the documents into clusters that correspond to individual or a particular set of labels. If we use these clustering information in our text classification, we can provide a far better result than conventional classifiers that are designed for good performance for binary or multi-class data sets.

In practice, very limited amount of labeled data may be available for training the classifiers. If the method is unsupervised, it does not take into account this labeling information which may prove valuable in enhancing the result. On the other hand, if the method is totally supervised, then the unlabeled data is considered useless. Only in semi-supervised approaches, both labeled and unlabeled data contribute in training.

But there are very few text classification approaches that are semi-supervised [10]. There are text classification approaches that consider its high dimensionality, some consider its multi-labelity and some try to train using a semi-supervised approach. But considering all of them together is rare. And solving all of these problems simultaneously, not just one of them, is the goal of this paper. Because, a system that does not consider one of these aspects will fail to provide the user with satisfactory results in practice. For example, *K Means Entropy* based method [8] uses subspace clustering method that is based on entropy of the dimensions. If the data is multi-label, then their entropy calculation no longer holds ground. Similarly, methods that are supervised depend heavily on the amount of labeled data and smaller amount of labeled data may hinder the generation of high quality classifiers.

In this paper, we propose a new subspace clustering technique that is in the later stage used in κ-nearest neighbor approach for classification of multi-label text data. The novelty in this subspace clustering approach is the application of *Impurity* component in measuring the cluster dispersions as well as the chi square statistic value for the dimensions. Using these two measures in our subspace clustering make it into a supervised approach as opposed to the legacy clustering approaches which are unsupervised. In order to use the unlabeled data in our training process, we performed a simple modification to our subspace clustering approach to make it a semi-supervised method.

After performing the subspace clustering, we move to our κ-NN approach where the neighbors of a test point (i.e. document) are the subspace cluster centroids trained on the training set. Based on these neighbors, the test point

is assigned a set of labels. These predicted labels of the test point are also ranked according to their probability of being present in the neighbors. We have applied other subspace clustering approaches for classification and our method provides significantly better results compared to them.

The contribution of this paper is three fold. First, we provide a semi-supervised subspace clustering algorithm called *SISC (Semi-supervised Impurity based Subspace Clustering)* that performs well in practice even when a very limited amount of labeled training data is available. Second, our subspace clustering algorithm successfully finds clusters in the subspace of dimensions even when the data is multi-label. To the best of our knowledge, this is the first attempt to classify multi-labeled documents using subspace clustering. Third, at the same time, this algorithm minimizes the effect of high dimensionality on the training. Finally, we compare *SISC* with other approaches to show the effectiveness of our algorithm over a number of data sets including data sets that are multi-labeled.

The organization of the paper is as follows: Section II discusses related works. Section III presents the theoretical background of our basic subspace clustering approach in supervised form. Section IV discusses the semi-supervised formulation of *SISC*. Section V, then provides the modification of our subspace clustering approach to handle multi labeled data. Section VI discusses the data sets, experimental setup and evaluation of our approach. Finally, Section VII concludes with directions to future work.

II. Related Work

Classifying text data has been an active area of research for a long time. Some of these research focus on some specific properties of text data. One such property is its multi-labelity. Multi-label classification studies the problem in which a data instance can have multiple labels. Approaches that have been proposed to address multi-label text classification, including margin-based methods, structural SVMs [14], parametric mixture models [16], κ-nearest neighbors (κ-NN) [19], and ensemble pruned methods [11]. One of the most recent works include *RAndom k-labELsets (RAKEL)* [15]. In a nutshell, it constructs an ensemble of LP classifiers and each LP is trained using a different small random subset of the multi-label set. Then, ensemble combination is achieved by thresholding the average zero-one decisions of each model per considered label. *MetaLabeler* is another approach which tries to predict the number of labels using SVM as the underlying classifier. Most of these methods utilize the relationship between multiple labels for collective inference. One characteristic of these models is they are mostly supervised [11], [13], [15].

Semi-supervised methods for classification is also present in the literature. This approach stems from the possibility of having both labeled and unlabeled data in the data set and in an effort to use both of them in training. In [3], Bilenko et al. propose a semi-supervised clustering algorithm derived from *K-Means*, *MPCK-MEANS*, that incorporates both metric learning and the use of pairwise constraints in a principled manner. There have also been attempts to find a low-dimensional subspace shared among multiple labels [8]. In [18], Yu et al. introduce a supervised *Latent Semantic Indexing (LSI)* method called *Multi-label informed Latent Semantic Indexing (MLSI)*. *MLSI* maps the input features into a new feature space that retains the information of original inputs and meanwhile captures the dependency of output dimensions. Our method is different from this algorithm as our approach tries to find clusters in the subspace. Due to the high dimensionality of feature space in text documents, considering a subset of weighted features for a class is more meaningful than combining the features to map them to lower dimensions [8]. In [4] a method called *LPI* is proposed. *LPI* is different from *LSI* which aims to discover the global Euclidean structure whereas *LPI* aims to discover the local geometrical structure. But *LPI* only handles multi-class data, not multi-label data. In [12] must-links and cannot-links, based on the labeled data, are incorporated in clustering. But, if the data is multi-label, then the calculation of must-link and cannot-link becomes infeasible as there are large number of class combinations and the number of documents in each of these combinations may be very low. As a result, this framework can not perform well when using multi-label text data.

There has been some subspace clustering approaches to minimize the impact of high dimensionality on classification. Subspace clustering can be divided into hard and soft subspace clustering. In case of hard subspace clustering, an exact subset of dimensions are discovered whereas soft subspace clustering determines the subsets of dimensions according to the contributions of the dimensions in discovering corresponding clusters. Examples of hard subspace clustering include *CLIQUE* [2], *PROCLUS* [1], *ENCLUS* [5] and *MAFIA* [7]. A hierarchical subspace clustering approach with automatic relevant dimension selection, called *HARP*, was presented by Yip et al. [17]. *HARP* is based on the assumption that two objects are likely to belong to the same cluster if they are very similar to each other along many dimensions. But, in multi-label and high dimensional text environment, the accuracy of *HARP* may drop as the basic assumption becomes less valid. In [9], a subspace clustering method called *nCluster* is proposed. But, it has similar problems when dealing with multi-label data.

Our algorithm uses subspace clusterinng and κ nearest neighbor approach. In this light, our work is more closely related with the work of Masud et al. [10]. In [10], a semi-supervised clustering approach called *SmSCluster* is used. This algorithm is specifically designed to handle evolving data streams. Although our multi-label text classification task is different in this perspective, we have used and

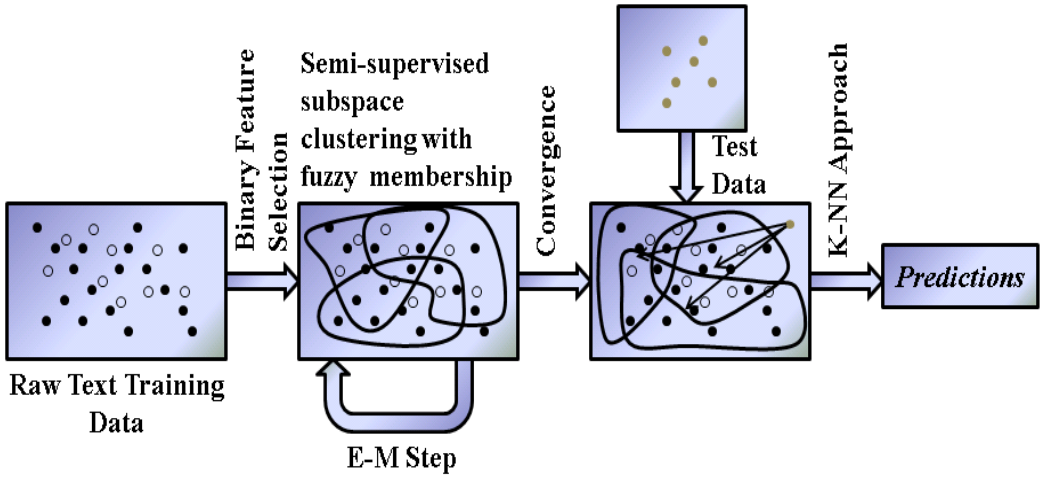


Figure 1. SISC Top Level Diagram

extended the cluster impurity measure used in *SmSCluster*. Also, *SmSCluster* is not designed to handle multi-labeled or high dimensional text data. Another closely related work to ours is the work of Jing et at. [8] and Frigui et at. [6]. The closeness is due the fuzzy and subspace clustering framework. But none of these works can perform better than our algorithm when the data is high dimensional and multi-labeled text data. The main reason behind this is our use of a novel subspace clustering algorithm that finds clusters in the high dimensional space and the fuzzy cluster membership that allows multiple labels to be effectively associated with a test document.

III. Impurity Based Subspace Clustering

A. Top Level Description

The semi-supervised clustering is based on the Expectation-Maximization(E-M) algorithm that locally minimizes an objective function. We use fuzzy clustering, allowing each data point to belong to multiple clusters. Since, in case of high dimensional text data, clusters can form in different subset of dimensions. We consider the weight of a dimension in a cluster to represent the probability of contribution of that dimension in forming the cluster. Then, we extract the summary statistics from the data points of each cluster. The progress of the algorithm can be partitioned into the following steps as shown in Figure 1.

1) *E-Step:* In the E-Step, the dimension weights and the cluster membership values are updated. The subspace clustering formulation is fuzzy in nature. So, each point can be a member of multiple clusters with different weights. Initially, every point, both labeled and unlabeled, is regarded as a member of all the clusters with equal weights. All the dimensions are also given equal weights.

2) *M-Step:* In this step, the centroids of the clusters are updated and the summary statistics, i.e. the representation of each class present in the cluster, are updated for use in the next step. During the summary calculation, the membership weights are summed up rather than using a threshold value to decide which point is regarded as a member of the cluster. We employ this approach so that membership weights can play useful role in class representation within a cluster.

3) *κ-NN formulation:* In this step, the κ nearest neighbor clusters are identified for each test point where κ is a user defined parameter. The distance is calculated in the subspace where the cluster resides. If κ is greater than 1, then during the class probability calculation, we multiply the class representation with the inverse of the distance and then sum them for each class across all the κ nearest clusters.

B. Subspace Clustering

We propose the following objective function to be used in our subspace clustering process by including the chi square statistic in our objective function. This component has been included in the objective function so that we can simultaneously minimize the within cluster dispersion and maximize the between cluster subspace distance to stimulate more dimensions to play an active role in the clustering process. Another component called *Impurity* [10] has been introduced to qualify the dispersion measure for each cluster. This component helps in generating purer clusters in terms of cluster labels.

The new objective function is written as follows:

$$F(W,Z,\Lambda)=\sum_{l=1}^{k}\left[\sum_{j=1}^{n}\sum_{i=1}^{m}w_{lj}^{f}\lambda_{li}^{q}D_{lij}*Impl+\gamma\sum_{i=1}^{m}\lambda_{li}^{q}\chi_{li}^{2}\right] \quad (1)$$

where

$$D_{lij}=(z_{li}-x_{ji})^2$$

subject to

$$\sum_{l=1}^{k}w_{lj}=1, 1\leq j\leq n, 1\leq l\leq k, w_{lj}\in(0,1)$$

$$\sum_{i=1}^{m} \lambda_{li} = 1, 1 \leq i \leq m, 1 \leq l \leq k, 0 \leq \lambda_{li} \leq 1$$

In this objective function, the parameter f controls the fuzziness of the membership of each data point, q further qualifies the weight of each dimension of each cluster λ_{li} and finally, γ controls the strength of the incentive given to the chi square component and dimension weights.

C. Impurity Measure

In the objective function in Eqn. 1, Imp_l is defined as

$$Imp_l = ADC_l * Ent_l$$

Here, ADC_l indicates the *Aggregated Dissimilarity Count* of cluster l and Ent_l denotes the entropy of cluster l. In order to understand ADC_l, we first need to define *Dissimilarity count* [10], $DC_l(x, y)$:

$$DC_l(x, y) = |L_l| - |L_l(c)|$$

if x is labeled and its label $y = c$, otherwise its value is 0. Then ADC_l becomes

$$ADC_l = \sum_{x \in L_l} DC_l(x, y)$$

where L_l indicates the set of labeled points in cluster l. The Entropy of a cluster l is computed as : $Ent_l = \sum_{c=1}^{C}(-p_c^l * log(p_c^l))$, where p_c^l is the prior probability of class c, i.e., $p_c^l = \frac{|L_l(c)|}{|L_l|}$.

We can show that ADC_l is proportional to the *gini index* of cluster l, $Gini_l$:

$$\begin{aligned} ADC_l &= \sum_{c=1}^{C}(|L_l(c)|)(|L_l| - |L_l(c)|) \\ &= (|L_l|)^2 \sum_{c=1}^{C}(p_c^l)(1 - p_c^l) \\ &= (|L_l|)^2(1 - \sum_{c=1}^{C}(p_c^l)^2) \\ &= (|L_l|)^2 * Gini_l \end{aligned}$$

This is the generalized version of calculation of ADC_l. But, we are considering fuzzy membership in our subspace clustering formulation. So, we have modified our ADC_l calculation. Rather than using counts, we use the membership weight for the calculation. This is reflected in the probability calculation.

$$p_c^l = \sum_{j=1}^{n} w_{lj} * j_c \tag{2}$$

where, j_c is 1 if data point j is a member of class c, 0 otherwise. This *Impurity Measure* is normalized using the global impurity measure, i.e. the impurity measure of the whole data set, before using in the subspace clustering formulation.

D. Chi Square Statistic

We define chi square component similar to conventional definition for our problem,

$$\chi_{li}^2 = \frac{n(ad - bc)^2}{(a + c)(b + d)(a + b)(c + d)}$$

where

$a = number\ of\ times\ feature\ i\ occurs\ in\ cluster\ l$
$b = number\ of\ times\ feature\ i\ occurs\ in\ all\ clesters\ except\ l$
$c = number\ of\ times\ cluster\ l\ occurs\ without\ feature\ i$
$d = number\ of\ times\ all\ clusters\ except\ l\ occur\ without\ feature\ i$
$n = number\ of\ dimensions$

Since we are using fuzzy cluster membership, a point can be member of multiple clusters. Therefore, if we try to calculate a, b, c, d and n then, we have to use a threshold to determine which point can be regarded as a member of a cluster. This not only brings forth another parameter, the membership values are undermined in the calculation. So, we modify the calculation of these counts by considering the corresponding membership values of each point. So, we get,

$$a = \sum_{j=1}^{n}\sum_{i \in j} w_{lj}, \quad b = 1 - \sum_{j=1}^{n}\sum_{i \in j} w_{lj}$$
$$c = \sum_{j=1}^{n}\sum_{i \notin j} w_{lj}, \quad d = 1 - \sum_{j=1}^{n}\sum_{i \notin j} w_{lj}$$
$$n = total\ number\ of\ labeled\ points$$

Since, for each individual point, the sum of membership values for different clusters in 1, the value of n is always the total number of labeled training points. The chi square component allows more features to be used during the clustering process thereby minimizing the effect of high and sparse dimensionality of the data.

E. Update Equations

Minimization of F in Eqn. 1 with the constraints forms a class of constrained nonlinear optimization problems. This optimization problem can be solved using partial optimization for Λ, Z and W. In this method, we first fix Z and Λ and minimize the reduced F with respect to W. Second, we fix W and Λ and minimize the reduced F with respect to Z. And finally, we minimize F with respect to Λ after fixing W and Z.

1) Dimension Weight Update Equation: Given matrices W and Z are fixed, F is minimized if

$$\lambda_{li} = \frac{1}{M_{lij} \sum_{i=1}^{m} \frac{1}{M_{lij}}} \tag{3}$$

where

$$M_{lij} = \left\{ \sum_{j=1}^{n} w_{lj}^f D_{lij} * Imp_l + \gamma \chi_{li}^2 \right\}^{\frac{1}{q-1}}$$

In order to get the above equation, first, we use the *Lagrangian Multiplier* technique to obtain the following

unconstrained minimization problem:

$$minF_1(\{\lambda_{li}\},\{\delta_l\}) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^f \lambda_{li}^q D_{lij} * Imp_l + \gamma \sum_{l=1}^{k}\sum_{i=1}^{m} \lambda_{li}^q \chi_{li}^2 - \sum_{l=1}^{k}\delta_l(\sum_{i=1}^{m}\lambda_{li} - 1) \quad (4)$$

where $[\delta_1, ..., \delta_k]$ is a vector containing the Lagrange Multipliers corresponding to the constraints. The optimization problem in Eqn. 4 can be decomposed into *k* independent minimization problems:

$$minF_{1l}(\lambda_{li},\delta_l) = \sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^f \lambda_{li}^q D_{lij} * Imp_l + \gamma \sum_{i=1}^{m} \lambda_{li}^q \chi_{li}^2 - \delta_l(\sum_{i=1}^{m}\lambda_{li} - 1) \quad (5)$$

for $l = 1, ..., k$. By setting the gradient of F_{1l} with respect to λ_{li} and δ_l to zero, we obtain

$$\frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{i=1}^{m}\lambda_{li} - 1\right) \quad (6)$$

$$= 0 \quad (7)$$

and

$$\frac{\partial F_{1l}}{\partial \lambda_{lt}} = \sum_{j=1}^{n} w_{lj}^f q \lambda_{lt}^{(q-1)} D_{lij} * Imp_l + \gamma q \lambda_{lt}^{(q-1)} \chi_{lt}^2 - \delta_l \quad (8)$$

$$= 0 \quad (9)$$

From Eqn. 9, we obtain

$$\lambda_{lt} = \frac{\delta_l^{\frac{1}{(q-1)}}}{\left[q\left\{\sum_{j=1}^{n} w_{lj}^f D_{lij} * Imp_l + \gamma\chi_{lt}^2\right\}\right]^{\frac{1}{(q-1)}}} \quad (10)$$

Substituting Eqn. 10 in Eqn. 7, we have

$$\sum_{i=1}^{m}\lambda_{li} = \delta_l^{\frac{1}{(q-1)}} \sum_{i=1}^{m} \frac{1}{\left[q\left\{\sum_{j=1}^{n} w_{lj}^f D_{lij} * Imp_l + \gamma\chi_{lt}^2\right\}\right]^{\frac{1}{(q-1)}}} = 1 \quad (11)$$

It follows that

$$\delta_l^{\frac{1}{(q-1)}} = \frac{1}{\sum_{i=1}^{m} \frac{1}{\left[q\left\{\sum_{j=1}^{n} w_{lj}^f D_{lij} * Imp_l + \gamma\chi_{lt}^2\right\}\right]^{\frac{1}{(q-1)}}}} \quad (12)$$

Substituting this expression back into Eqn. 10, we get

$$\lambda_{li} = \frac{1}{M_{lij}\sum_{i=1}^{m}\frac{1}{M_{lij}}}$$

where

$$M_{lij} = \left\{\sum_{j=1}^{n} w_{lj}^f D_{lij} * Imp_l + \gamma\chi_{li}^2\right\}^{\frac{1}{q-1}}$$

2) Cluster Membership Update Equation: Similar to the dimension update equation, we can derive the update equations for cluster membership matrix i.e. *W*, given *Z* and Λ are fixed. The update equations are as follows:

$$w_{lj} = \frac{1}{N_{lij}\sum_{l=1}^{k}\frac{1}{N_{lij}}} \quad (13)$$

where

$$N_{lij} = \left\{\sum_{i=1}^{m}\lambda_{li}^q D_{lij}\right\}^{\frac{1}{f-1}}$$

In order to derive the above equation, similar to the dimension update formulation, we use the *Lagrangian Multiplier* technique to obtain an unconstrained minimization problem and by setting the gradient of F_{1l} with respect to w_{lj} and δ_l to zero, we obtain

$$\frac{\partial F_{1l}}{\partial \delta_l} = \left(\sum_{l=1}^{k} w_{lj} - 1\right) = 0 \quad (14)$$

and

$$\frac{\partial F_{1l}}{\partial w_{lt}} = \sum_{i=1}^{m} f w_{lt}^{(f-1)} \lambda_{li}^q D_{lij} * Imp_l - \delta_l = 0 \quad (15)$$

From Eqn. 15, we obtain

$$w_{lt} = \frac{\delta_l^{\frac{1}{(f-1)}}}{\left[f\left\{\sum_{i=1}^{m}\lambda_{li}^q D_{lij} * Imp_l\right\}\right]^{\frac{1}{(f-1)}}} \quad (16)$$

From here, we can derive

$$w_{lj} = \frac{1}{N_{lij}\sum_{l=1}^{k}\frac{1}{N_{lij}}}$$

where

$$N_{lij} = \left\{\sum_{i=1}^{m}\lambda_{li}^q D_{lij}\right\}^{\frac{1}{f-1}}$$

3) Cluster Centroid Update Equation: The cluster center update formulation is similar to the formulation of dimension and membership update equations. We can derive the update equations for cluster center matrix i.e. *Z*, given *W* and Λ are fixed. The update equation is as follows:

$$z_{li} = \frac{\sum_{j=1}^{n} w_{lj}^f x_{ij}}{\sum_{j=1}^{n} w_{lj}^f} \quad (17)$$

IV. Semi-supervised Impurity Based Subspace Clustering For Multi Class Data

Multiplying the impurity with the dispersion in the objective function in Eqn. 1 makes the classification fully supervised. If there is unlabeled data present, then we can consider them by adding a dispersion component in the objective function without the impurity factor. The new objective function, therefore, becomes

$$F(W,Z,\Lambda) = \sum_{l=1}^{k}\sum_{j=1}^{n}\sum_{i=1}^{m} w_{lj}^{f}\lambda_{li}^{q}D_{lij} * (1+Imp_l) + \sum_{l=1}^{k}\sum_{i=1}^{m}\lambda_{li}^{q}\chi_{li}^{2} \quad (18)$$

As a result of this change, the dimension update equation, i.e. Eqn. 19 also changes. The new dimension weight update equation is as follows:

$$\lambda_{li} = \frac{1}{M_{lij}\sum_{i=1}^{m}\frac{1}{M_{lij}}} \quad (19)$$

where

$$M_{lij} = \left\{\sum_{j=1}^{n} w_{lj}^{f}D_{lij} * (1+Imp_l) + \gamma\chi_{li}^{2}\right\}^{\frac{1}{q-1}}$$

The other update equations remain same as they are independent of the *Impurity* component.

V. Semi-supervised Impurity Based Subspace Clustering For Multi Labeled Data

If the data is multi-labeled, the impurity measure in the previous section is not correct. As the classes may overlap, the probability calculation becomes incorrect, i.e., the sum of probabilities may become greater than 1. We, therefore, modify the impurity calculation in the generalized case (i.e. not fuzzy) as follows:

The Entropy of a cluster l is computed as : $Ent_l = \sum_{c=1}^{C}(-p_c^l * log(p_c^l) - (1-p_c^l) * log(1-p_c^l))$, where p_c^l is the prior probability of class c, i.e., $p_c^l = \frac{|L_l(c)|}{|L_l|}$.

We modify ADC_l and we can show that ADC_l is proportional to the multi label *gini index* of cluster l:

$$\begin{aligned} ADC_l &= \sum_{x\in L_l}\left(DC_l(x,y) + DC_l'(x,y)\right) \\ &= \sum_{c=1}^{C}((|L_l(c)|)(|L_l| - |L_l(c)|) + (|L_l(c')|)(|L_l| - |L_l(c')|)) \\ &= (|L_l|)^2\sum_{c=1}^{C}((p_c^l)(1-p_c^l) + (p_c'^l)(1-p_c'^l)) \\ &= (|L_l|)^2(C - \sum_{c=1}^{C}(p_c^l)^2 - \sum_{c=1}^{C}(1-p_c^l)^2) \\ &= (|L_l|)^2 * Gini_l \end{aligned}$$

where, c' consists of all classes except c and $Gini_l$ is the *gini index* for multi-labeled data.

We can then use this ADC_l in our calculation of *Impurity*. It is apparent that, all the update equations remain the same, only the calculation of *Impurity* differs. We apply the previous formulation of fuzzy probability calculation in Eqn. 2 in this case too, in order to use the multi-label impurity measure in our model.

VI. Experiments and Results

We have performed extensive experiments to find out the performance of our method in both multi-class and multi-label environment. In the next part, we will describe the data sets used in the experiments and also the base line methods against which we have compared our results.

A. Data Sets

We have used a number of datasets in our experimentation. In this paper, due to space constraints, we have reported only 4 of those data sets. Two of these datasets are multi-class datasets and the other two are multi-label datasets. In all cases, we used fifty percent data as training and rest as test data in our experiments as part of 2-fold cross-validation. Similar to other text classification approaches, we performed preprocessing the data and removed stop words from the data. We used binary features as dimensions, i.e. features can only have 0 or 1 values. The parameter γ is set to 0.5. For convenience, we selected 1000 features based on information gain and used them in our experiments. In all the experiments, the same feature set was used. We performed multiple runs on our data sets with the training set chosen randomly from the data set. The four data sets used are as follows:

1) Reuters Data Set: This is part of the Reuters-21578, Distribution 1.0. We selected 10,000 data points from the 21,578 data points of this data set and henceforth, this part of the data set will be referred to as simply Reuters Data Set. We considered the most frequently occurring 20 classes in our experiments. Of the 10,000 data points, 6651 are multi-labeled. This data set, therefore, allows us to determine the performance of our multi-label formulation.

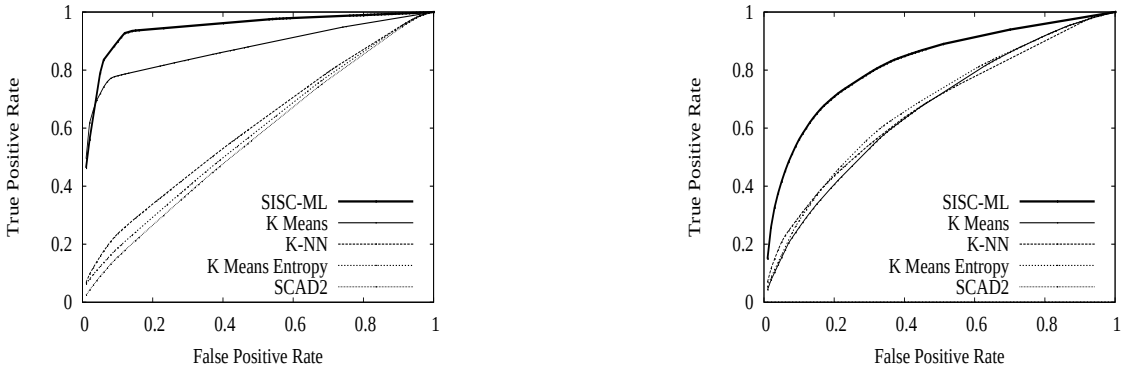


Figure 2. (a) ROC Curves For NSF Abstracts Data Set. (b) ROC Curves For 20 Newsgroups Data Set Without Multi-Labels.

Methods	NSF Abstracts	20 Newsgroups w/o multi-label
SISC multi-class	**0.944**	**0.84**
K Means	0.869	0.661
κ-NN	0.602	0.666
SCAD2	0.56	0.661
K Means Entropy	0.58	0.68

Table I
AREA UNDER THE ROC CURVE COMPARISON CHART FOR MULTI-CLASS CLASSIFICATION

2) 20 Newsgroups Data Set: This data set is also multi-label in nature. We selected 15,000 thousand documents randomly for our classification experiments. Of them 2822 are multi-label documents and the rest are single labeled. We have performed our classification on the top 20 classes of this data set.
3) NSF Abstracts Data Set: This is a multi-class data set. Each document or abstract is associated with a single area of research. The classes indicate the area of research. The total number of documents is 1,34,158. From them, 10,000 documents are randomly selected to represent the top 10 classes in the data set. We have used this reduced set in our experiments.
4) 20 Newsgroups Data Set Without Multi-Labels: We have removed the multi-label documents from the previously mentioned data set to create this multi-class data set. The number of labels considered is the same as the multi-label data set,i.e. 20 classes. In this case, we selected 12,000 documents randomly and used them in our experiments.

B. Base Line Approaches

We have 3 parts in our experiments. In the first phase, we show comparison with basic K means clustering and κ-nearest neighbor (κ-NN) approach. In the second phase, we show the comparison between different subspace clustering approach and our method. Finally, we show the performance of our proposed approach in comparison with two multi-label classification approaches. All the comparisons are done based on ROC curves i.e. the area under the curve. This area can have a range from 0 to 1.

1) Basic κ-NN Approach and K Means Clustering: In this part, we compare our approach with the basic *κ-NN* approach and *K Means Clustering*. in *κ-NN* approach, all the training data points are saved and based on user specified parameter κ, we find the nearest κ neighboring data points of a test instance and based on their labels, we decide on the classification of that test instance. In the *K Means Clustering* approach, the data points are divided into different clusters. In order to use *K Means Clustering* for classification, after performing the clustering, we find the κ nearest clusters and based on the distribution of labels in those clusters, we predict the labels for a test instance. A similar method has been applied in [10], however, we are not dealing with data streams in this case. So, we train a single classifier model and perform the test with that model as opposed to training multiple models on different data chunks in an ensemble fashion [10]. We use this approach as baseline because of the high dimensionality of text data. If only single points are used for nearest neighbors, the features present in them may not provide us with correct classification information. There may also be cases where there are no appropriate nearest neighbors because few features coincide in both test and training instance. Also, using this method as baseline allows us to show the effectiveness of introducing subspace clustering in our approach.

2) Subspace Clustering Methods: Introducing subspace clustering is not enough to perform good classification in the text data. We have to consider the high dimensionality and label information in our subspace clustering formulation. In this section, we show how our subspace clustering approach performs better in text classification than some state of the art subspace clustering algorithms. We provide comparison with *SCAD2* [6] and *K Means Entropy* [8] based approach which indicates the significantly better classification performance of our approach. The reason for choosing *SCAD2*

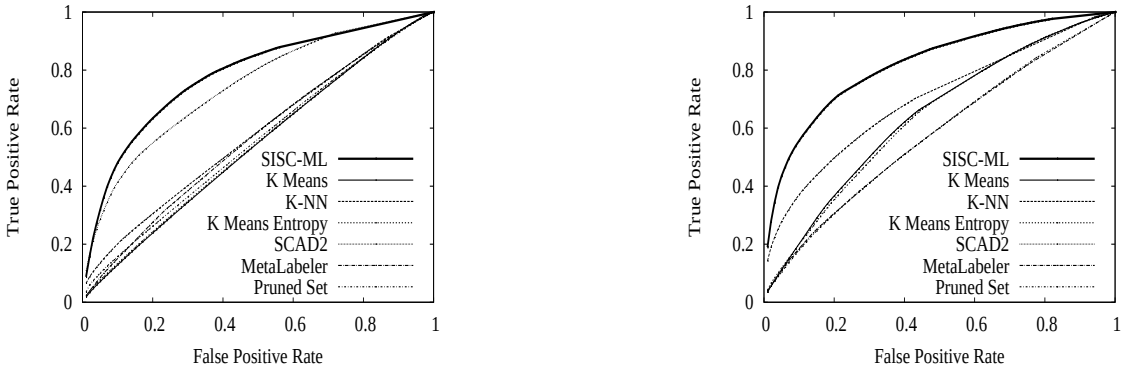


Figure 3. (a) ROC Curves For Reuters Data Set. (b) ROC Curves For 20 Newsgroups Data Set.

Methods	Reuters	20 Newsgroups
SISC multi-Label	**0.78**	**0.82**
Pruned Set	0.55	0.58
MetaLabeler	0.564	0.578
K Means	0.539	0.642
κ-NN	0.577	0.698
SCAD2	0.742	0.642
K Means Entropy	0.542	0.638

Table II
AREA UNDER THE ROC CURVE COMPARISON CHART FOR MULTI-LABEL CLASSIFICATION

is because of its close resemblance to our algorithm, to show the effect of introducing impurity measure and chi square component in the subspace clustering formulation and because, like ours, *SCAD2* is also fuzzy in nature.

3) Multi-Label Classification: To show the feasibility of the multi-label variation of our algorithm, we compare it with two multi-label classification algorithms, the *Pruned Set* algorithm [8] and *MetaLabeler* [13]. In both cases, we used *SVM* as the underlying classifier as used in their algorithm. Only the multi-label data sets mentioned above were used for this part of the experiments.

In the *Pruned Set* [11] method, based on a user specified parameter, all data points with label combinations having sufficient count are added to an empty training set. This training set is then augmented with rejected data points having label combinations that are not sufficiently frequent. This is done by making multiple copies of the data points, only this time with subsets of the original label set. So, some data points may be duplicated during this training set generation process. This training set is then used to create an ensemble of *SVM* classifiers. We have also varied the number of retained label subsets to add to the training set and chose the best result to report.

In case of *MetaLabeler* [13], there are two sets of classifiers. One set consists of binary classifiers that correspond to each of the unique labels present in the data set. The other set consists of a single multi-class classifier that learns the number of labels associated with each data point. In [13], the authors provide three strategies for training this classifier using SVM. We used the approach that produces the best results as claimed by the authors, i.e. using the feature set used in the first set of classifiers but the labels are the number of labels in the multi-label data points. Based on the predictions of this classifier, we choose the labels that have the maximum values in predictions found through the first set of classifiers mentioned above.

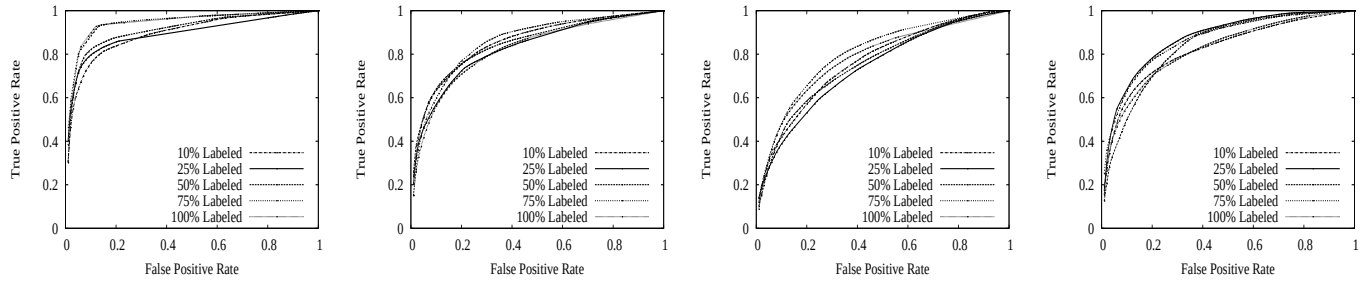
C. Discussion

In Figure 2(a), we compare the semi-supervised multi-class variation of *SISC* formulated in Section IV with base line approaches for the *NSF Abstracts Data Set*. As can be seen from the figure, our method provides significantly better result than other methods. Since, this data set is not multi-label, we do not show the performance of *Pruned Set* and *MetaLabeler* methods on this data set. Our algorithm achieves an AUC (Area Under The Curve) value of 0.944 whereas the closest any other method can achieve is 0.869.

In Figure 2(b), we perform the same comparison, but for the *20 Newsgroups Data Set Without Multi-Labels*. As can be seen from the figure, in this case too, our method provides significantly better result than other methods. Our algorithm achieves an AUC (Area Under The Curve) value of 0.84 whereas the closest any other method can achieve is 0.68.

In Table I, we present the summary of our results in terms of *AUC* values with a range from 0 to 1.

In Figure 3(a) and Figure 3(b), we show the performance of the multi-label variation of our algorithm. We have added the *Pruned Set* and *MetaLabeler* method in our comparison as they are state-of-the-art multi-label algorithms. Also, these two graphs represent the experimental results on Reuters and 20 Newsgroups Data Sets respectively, both of which are multi-label data sets. These two figures are followed by Table II, which summarizes our multi-label experiment results. For Reuters data set, our algorithm achieves AUC value of 0.78 and the nearest value is 0.742.



(a) (b) (c) (d)

Figure 4. ROC Curves For Different Percentage Of Labeled Data (a) NSF Abstracts Data Set. (b) 20 Newsgroups Data Set Without Multi-Labels (c) Reuters Data Set (d) 20 Newsgroups Data Set.

DataSets	10% Labeled Data	25% Labeled Data	50% Labeled Data	75% Labeled Data	100% Labeled Data
NSF Abstracts	0.894	0.891	0.911	0.944	0.944
20 Newsgroups w/o Multi-Label	0.853	0.826	0.847	0.856	0.82
Reuters	0.763	0.737	0.755	0.802	0.78
20 Newsgroups	0.82	0.871	0.835	0.865	0.82

Table III
AREA UNDER THE ROC CURVE COMPARISON CHART FOR DIFFERENT PERCENTAGE OF LABELED DATA

And, for 20 Newsgroups data set, our algorithm achieves AUC value of 0.82 whereas, the nearest value is 0.698.

D. Performance On Limited Labeled Data

We have varied the amount of labeled data in our data sets to find out how this aspect impacts the performance of our algorithm. As can be seen from Figure 4, even with significant change in the amount of labeled data, the performance of our algorithm is quite satisfactory. In case of Reuters and 20 Newsgroups Data Set, we use the multi-label variation of *SISC* and for the other two multi-class data sets, we use the multi-class variation of *SISC*. The AUC values are summarized in Table III.

VII. CONCLUSIONS

In this paper, we have presented *SISC*, a new semi-supervised text classification algorithm based on fuzzy subspace clustering approach. Our proposed subspace clustering algorithm identifies clusters in the subspace for high dimensional sparse data and and we then use them for classification using κ-*NN* approach. Also, our formulation of this fuzzy clustering allows us to handle multi-labeled text data. *SISC*, being semi-supervised, uses both labeled and unlabeled data during clustering process and as can be seen from empirical evaluation, performs well even when limited amount of labeled data is available. The experimental results on real world multi-class and multi-labeled data sets have shown that *SISC* outperforms κ-*NN*, *K Means Clustering*, *K Means Entropy* based method, *SCAD2* and state-of-the-art multi-label text classification approaches like *Pruned Set* and *MetaLabeler* in classifying text data. There are still scopes for improvement as well as possibility of extending this new algorithm. In future, we would like to incorporate label propagation in our classification approach for better classification model as well as train not only one but multiple classifiers in an ensemble model. We would also like to extend our algorithm to classify streaming text data.

REFERENCES

[1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28(2):61–72, 1999.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, 1998.

[3] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *In ICML*, pages 81–88, 2004.

[4] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *Knowledge and Data Engineering, IEEE Transactions on*, 17(12):1624–1637, Dec. 2005.

[5] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, New York, NY, USA, 1999. ACM.

[6] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567 – 581, 2004.

[7] S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. *Technical Report CPDC-TR-9906-010, Northwest Univ.*, 1999.

[8] L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.*, 19(8):1026–1041, 2007.

[9] G. Liu, J. Li, K. Sim, and L. Wong. Distance based subspace clustering with flexible dimension partitioning. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1250–1254, April 2007.

[10] M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 929–934, Dec. 2008.

[11] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 995–1000, Dec. 2008.

[12] J. Struyf and S. Džeroski. Clustering trees with instance level constraints. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 359–370, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 211–220, New York, NY, USA, 2009. ACM.

[14] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 104, New York, NY, USA, 2004. ACM.

[15] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.

[16] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. *In Advances in Neural Information Processing Systems 15. Cambridge: MIT Press.*, 2003.

[17] K. Yip, D. Cheung, and M. Ng. Harp: a practical projected clustering algorithm. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1387–1397, Nov. 2004.

[18] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM.

[19] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048, 2007.